Ouroboros Samasika

Chanel Richardson

Mina Blockchain Model

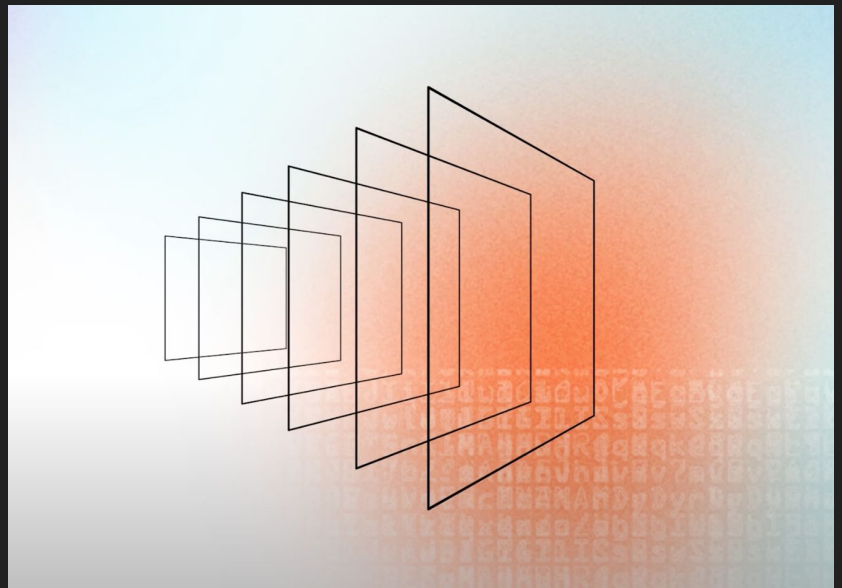
- Goal: Increase the decentralized aspect of cryptocurrency
- Succinct blockchain (entire blockchain is 22kB)
- Consensus Mechanism: Ouroboros Samasika

Mina Blockchain Definition

- An unambiguous usable representation (ie. not merely hashes) of the parts of the state a typical user would care about — namely the current balance of their account.
- The data that a node needs in order to verify that this state is real in a trustless manner.
- The ability to broadcast transactions on the network to make a transfer.

Succinct Blockchain: Replacing the Blockchain with a Blockchain Summary

- Each block produced in the Mina protocol contains a SNARK
- SNARK - “Succinct Non Interactive Argument of Knowledge”



Succinct Blockchain: Types of Participants

- Bitcoin: Full nodes, light nodes, mining nodes
- Mina: Full nodes, block producers, snark workers

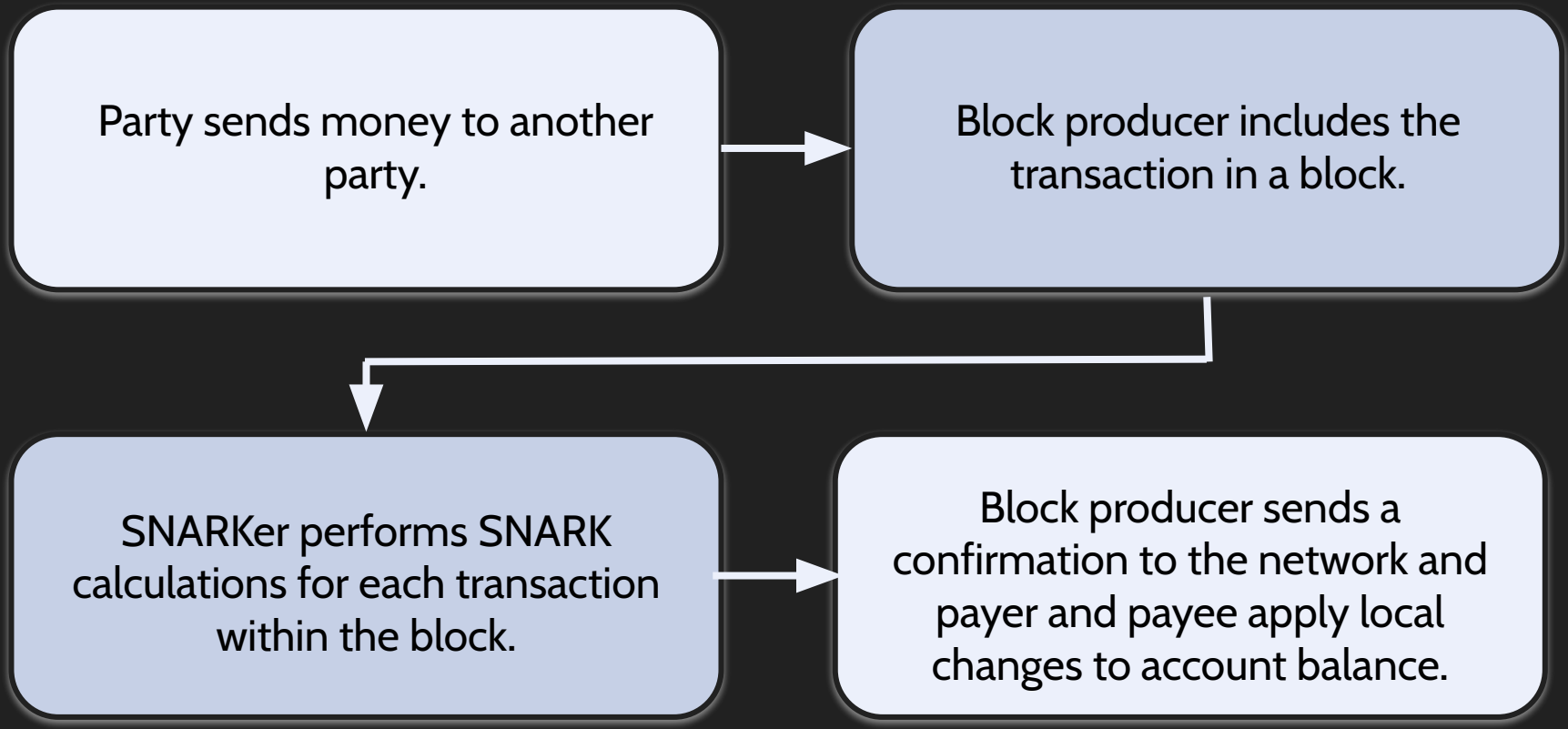
Succinct Blockchain: Types of Participants

Full Node
Uses SNARKs to validate transactions and chain history. Not a necessary part of Mina.

Block Producer
Produce new blocks, achieve consensus, and generate SNARK for the entire block. Keeps last 290 blocks

SNARKer
Create SNARKs for each individual transaction within a block. Paid by block producers. Keeps full blockchain.

Flow of a Payment in Mina Protocol



Ouroboros Samasika

Background: Ouroboros (Original, Praos, Genesis)

- The consensus protocol outlined in this paper Ouroboros Samasika is based on the Ouroboros family of consensus protocols
- Very similar to Snow White
- Protocols are provably secure proof of stake protocols in the synchronous and/or semi-synchronous settings
- Ouroboros Samasika is directly based on Ouroboros Genesis

Ouroboros Genesis Overview

- Dynamic Availability
- Allow newly joining or re-joining parties to get full view of the network without relying on other nodes
- “Bootstrap from Genesis”

Parties in the protocol

- Alert Parties
- Potentially Active Parties
- Inactive Parties

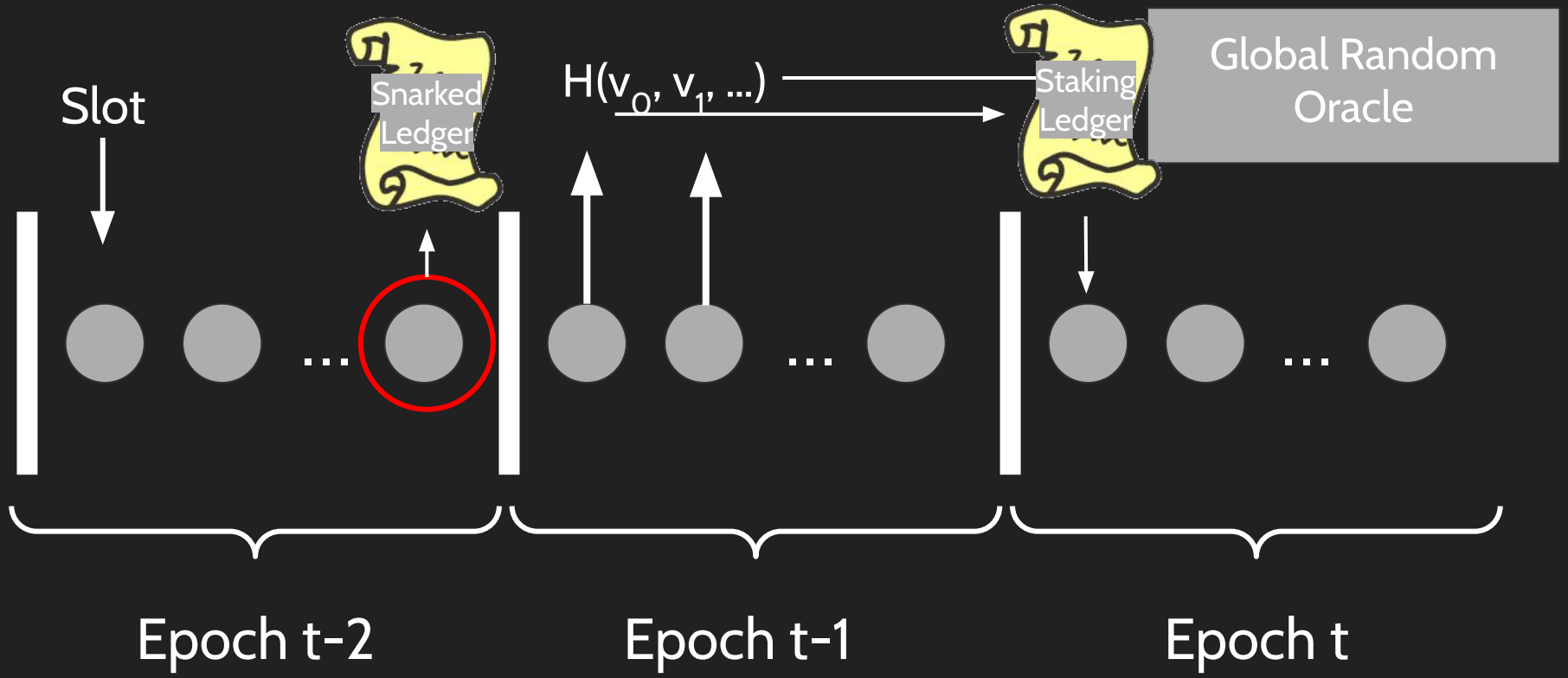
$$alert :\Leftrightarrow operational \wedge time\text{-}aware \wedge online \wedge synchronized$$

$$active :\Leftrightarrow (operational \wedge time\text{-}aware \wedge online) \vee adversarial \vee time\text{-}unaware$$

Party Registration

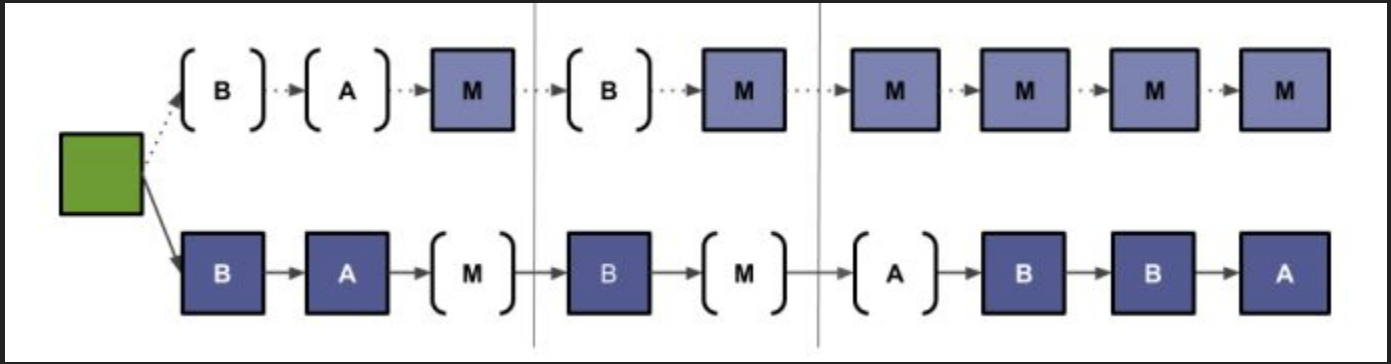
- Global Clock
- Global Random Oracle
- Synchronize to the network

Epochs and Slots

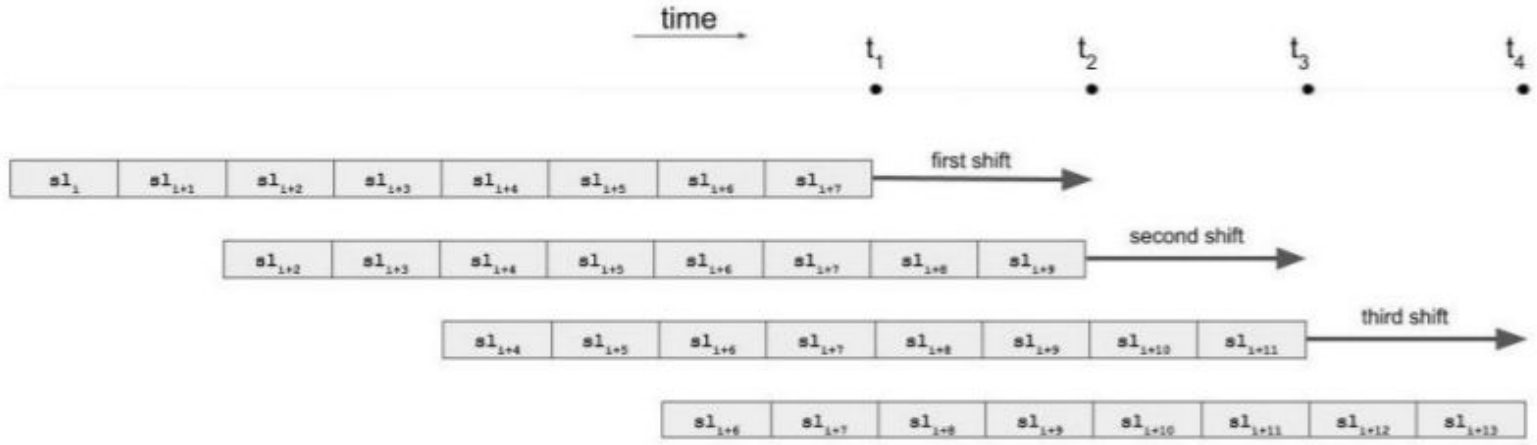


Dealing with Forks

- Longest chain selection in the case of a **short range fork**
- Finding the densest chain in the s slots following a fork in the case of a **long range fork** (generally due to long range attacks)
- “Plenitude Rule”

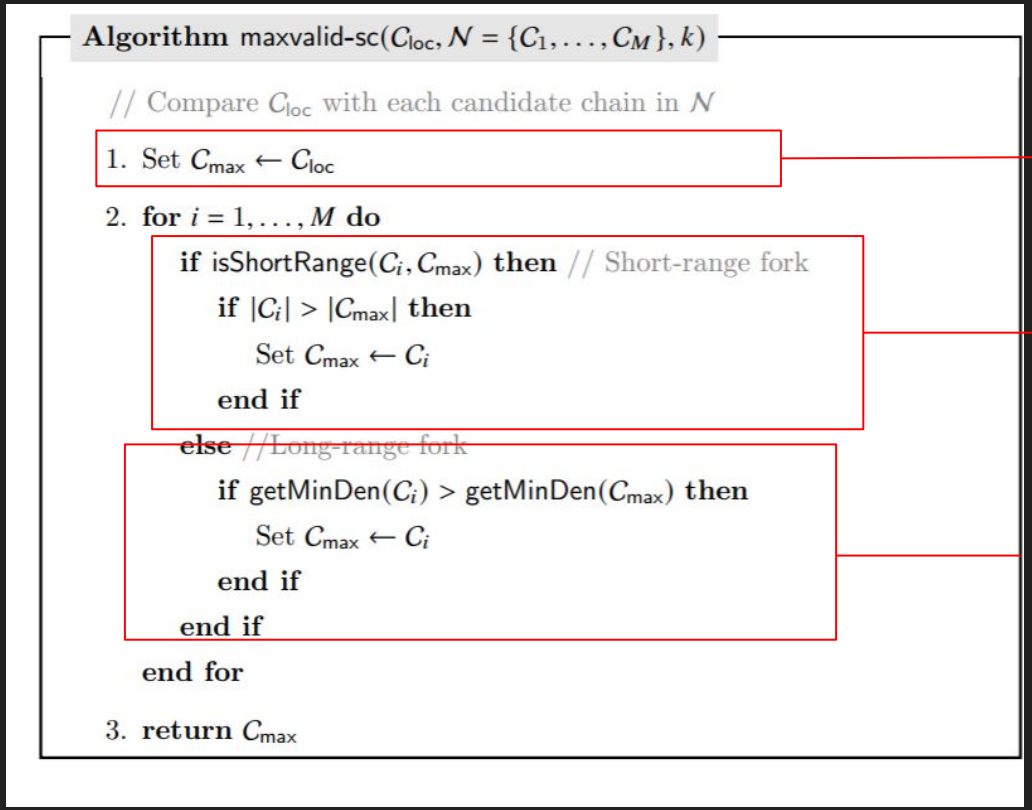


V-Shifting, ω-window



$$\omega = (1 + \epsilon_s)s_{CG}$$
$$V = \epsilon_s s_{CG}$$

Maxvalid-SC



Algorithm maxvalid-sc($C_{\text{loc}}, \mathcal{N} = \{C_1, \ldots, C_M\}, k$)

// Compare C_{loc} with each candidate chain in $\mathcal{N}$

1. Set $C_{\text{max}} \leftarrow C_{\text{loc}}$
2. **for** $i = 1, \ldots, M$ **do**
   - **if** isShortRange(C_i, C_{max}) **then** // Short-range fork
     - **if** $|C_i| > |C_{\text{max}}|$ **then**
       - Set $C_{\text{max}} \leftarrow C_i$
     - **end if**
   - **else** //Long-range fork
     - **if** getMinDen(C_i) > getMinDen(C_{max}) **then**
       - Set $C_{\text{max}} \leftarrow C_i$
     - **end if**
   - **end if**

   end for
3. **return** C_{max}

Set the local chain to the longest valid chain

Check whether there is a short range fork between the current chain and the compared chain, if so set the current max chain to the compared chain

If it's not a short range fork, check if it's a long range fork, and get the minimum density seen in any window in each chain. Adopt the chain with the higher minimum density as the new current max

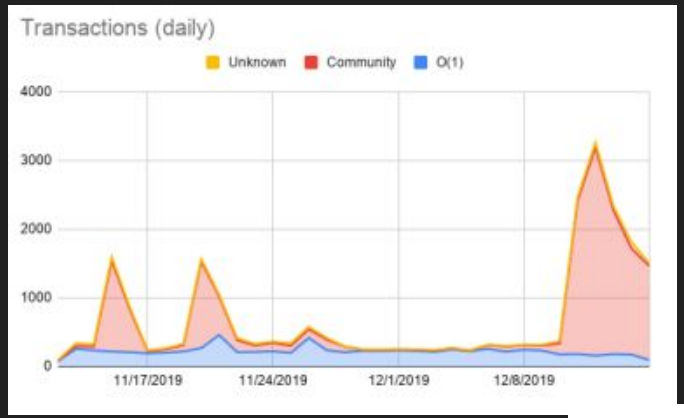
Security Guarantees

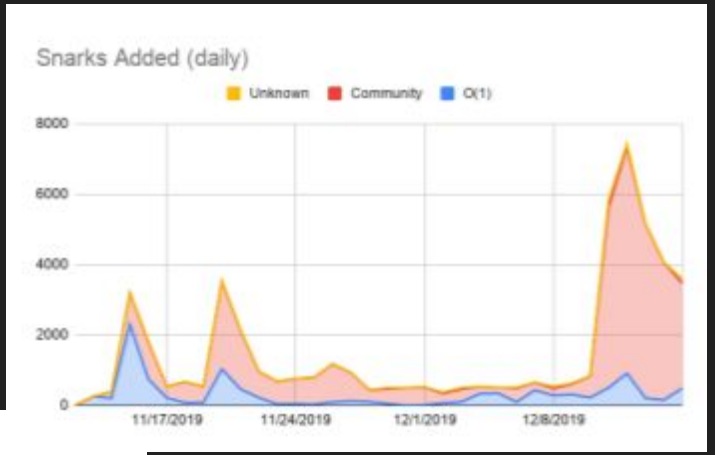
- Common Prefix: If 2 alert parties look at their chains at the onset of slots s_1 and s_2 where $s_1 \leq s_2$, the chain seen in s_1 with the exception of the last k blocks precedes the chain seen in s_2.
- Chain growth: For a slot, $s_1 + s \leq s_2$, the chain will have grown by at least $\tau * s$ blocks during that time, τ is the speed coefficient
- Chain quality: Any portion of the chain, k that an alert party sees at the beginning of a slot, has a ratio of at least μ blocks originating from alert parties, μ is the chain quality coefficient.

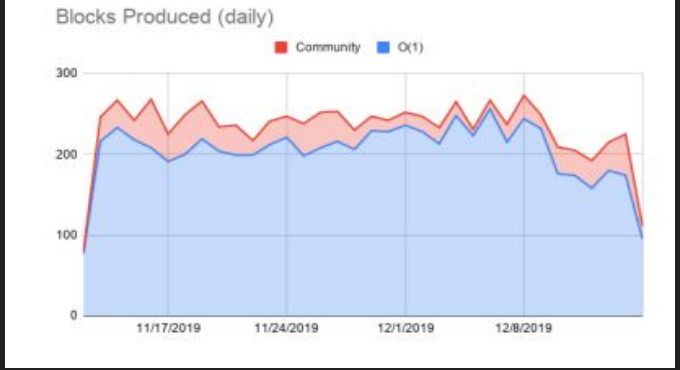
Lemmas 3, 4, and 5

- Shows that the least dense window in the selected chain has a number of blocks $> t_{high}$, with high probability
- Show that the least dense window in a non-selected chain has a number of blocks $< t_{low}$, with high probability
- Shows that $t_{high} > t_{low}$

Test Run of Ouroboros Samasika







Summary

- Mina protocol is a succinct blockchain protocol allowing full nodes to have all information needed for verification in 22kB
- Ouroboros Samasika builds off of Ouroboros Genesis to provide consensus in the succinct setting
- Chain selection is performed based on density of candidate chains after forking

Questions?